

An estimator for Poisson means whose relative error distribution is known

Mark Huber

Version: June 1, 2016

Abstract

Suppose that X_1, X_2, \dots are a stream of independent, identically distributed Poisson random variables with mean μ . This work presents a new estimate μ_k for μ with the property that the distribution of the relative error in the estimate $((\hat{\mu}_k/\mu) - 1)$ is known, and does not depend on μ in any way. This enables the construction of simple exact confidence intervals for the estimate, as well as a means of obtaining fast approximation algorithms for high dimensional integration using TPA. The new estimate requires a random number of Poisson draws, and so is best suited to Monte Carlo applications. As an example of such an application, the method is applied to obtain an exact confidence interval for the normalizing constant of the Ising model.

Keywords: randomized approximation scheme, high-dimensional integration, tpa

MSC 2010: 68W20, 62L12

1 Introduction

A random variable X is Poisson distributed with mean μ (write $X \sim \text{Pois}(\mu)$) if $\mathbb{P}(X = i) = \exp(-\mu)\mu^i/i!$ for $i \in \{0, 1, 2, \dots\}$. Suppose that X_1, X_2, \dots are independent identically distributed (iid) Poisson random variables with mean μ . The purpose of this paper is to present a new estimator for μ that uses almost the ideal number of Poisson draws.

Our estimate will not only use draws from $X_1, X_2, \dots \stackrel{\text{iid}}{\sim} \text{Pois}(\mu)$, but make extra random choices as well. This external source of randomness can

be represented by a random variable U that is uniformly distributed over $[0, 1]$ (write $U \sim \text{Unif}([0, 1])$). As is well known, a single draw U is equivalent to an infinite number of draws $U_1, U_2, \dots \stackrel{\text{iid}}{\sim} \text{Unif}([0, 1])$.

Definition 1. Suppose \mathcal{A} is a computable function of $X_1, X_2, \dots \stackrel{\text{iid}}{\sim} \text{Pois}(\mu)$ and auxiliary randomness (represented by $U \sim \text{Unif}([0, 1])$) that outputs $\hat{\mu}$. Let T be a stopping time with respect to the natural filtration so that the value of $\hat{\mu}$ only depends on U and X_1, \dots, X_T . Then call T the *running time* of the algorithm.

Definition 2. For an estimate $\hat{\mu}$ of μ , the *relative error* is $\epsilon_{\text{relative}} = (\hat{\mu}/\mu) - 1$. Call $\hat{\mu}$ an (ϵ, δ) -*approximation* for μ if $\mathbb{P}(|\epsilon_{\text{relative}}| > \epsilon) < \delta$.

The simplest algorithm for estimating μ just fixes $T = n$, and sets

$$\hat{\mu}_n = \frac{X_1 + \dots + X_n}{n}.$$

This basic estimate has several good properties. First, it is unbiased, that is, $\mathbb{E}[\hat{\mu}_n] = \mu$. Second, it is consistent, as $n \rightarrow \infty$, $\hat{\mu}_n \rightarrow \mu$ with probability 1. Third, it is efficient. Using the Fisher information about μ contained in a single X_i with the Crámer-Rao inequality, it is possible to show that this estimate has the minimum variance of any unbiased estimate that only uses n draws.

However, this estimate is difficult to use for building (ϵ, δ) -approximation algorithms, as the ratio $\hat{\mu}_n/\mu$ depends strongly on μ . It is well known that $X_1 + \dots + X_n \sim \text{Pois}(n\mu)$. Using techniques such as Chernoff bounds to bound the tail of a Poisson distribution, it is possible to bound the value of n needed to get an (ϵ, δ) -approximation.

These bounds however are not tight, and inevitably a slightly larger value of n than is necessary will be needed to meet the (ϵ, δ) requirements.

The goal of this work is to introduce a new estimate for the mean of the Poisson distribution whose relative error is independent of μ , the quantity being estimated.

1.1 Examples of estimates whose relative error is independent of the parameter

As an example of a distribution where the basic estimate is scalable, say that Z is normally distributed with mean μ and variance σ^2 (write $Z \sim \text{N}(\mu, \sigma^2)$)

if Z has density $f_Z(s) = (2\pi\sigma^2)^{-1/2} \exp(-(s - \mu)^2/[2\sigma^2])$. As is well known, normals can be scaled and shifted, and still remain normal.

Fact 1. For $Z \sim \mathbf{N}(\mu, \sigma^2)$ and constants a and b , $aZ + b \sim \mathbf{N}(a\mu + b, a^2\sigma^2)$.

Now consider $Z_1, Z_2, \dots \stackrel{\text{iid}}{\sim} \mathbf{N}(\mu, \mu^2)$. In this case, the sample average satisfies $\hat{\mu}_n \stackrel{\text{iid}}{\sim} \mathbf{N}(\mu, \mu^2/n)$, and $(\hat{\mu}_n/\mu) - 1 \sim \mathbf{N}(0, 1/n)$. Note that the distribution of the relative error does not depend in any way on the parameter μ being estimated.

For another example, say that Y is exponentially distributed with rate μ (write $Y \sim \text{Exp}(\mu)$) if Y has density $f_Y(s) = \mu \exp(-\mu s) \mathbf{1}(s \geq 0)$. Here $\mathbf{1}(\cdot)$ is the indicator function that is 1 when the argument inside is true, and 0 when it is false. As with normals, scaled exponentials are still exponential. Unlike normals, the rate parameter is divided by the scale.

Fact 2. For $Y \sim \text{Exp}(\mu)$ and constant a , $aY \sim \text{Exp}(\mu/a)$.

Say that T has a Gamma distribution with shape parameter k and rate parameter μ (write $T \sim \text{Gamma}(k, \mu)$) if T has density

$$f_T(s) = \mu^k \Gamma(k)^{-1} s^{k-1} e^{-\mu s} \mathbf{1}(s \geq 0).$$

Adding iid exponentially distributed random variables together gives a Gamma distributed random variable.

Fact 3. If $Y_1, Y_2, \dots \stackrel{\text{iid}}{\sim} \text{Exp}(\mu)$, then for any k , $Y_1 + \dots + Y_k \sim \text{Gamma}(k, \mu)$.

Given n draws Y_1, \dots, Y_n , the maximum likelihood estimator for μ in this context is the inverse of the sample average (see for instance [8]):

$$\hat{\mu}_{\text{MLE},n} = \frac{n}{Y_1 + \dots + Y_n}.$$

That gives

$$\frac{\hat{\mu}_{\text{MLE},n}}{\mu} = \frac{n}{\mu Y_1 + \dots + \mu Y_n}.$$

By scaling $\mu Y_1 \sim \text{Exp}(\mu/\mu) = \text{Exp}(1)$, so $\mu Y_1 + \dots + \mu Y_n \sim \text{Gamma}(n, 1)$. Therefore the relative error in $\hat{\mu}_{\text{MLE},n}$ is independent of μ !

Now, the distribution of $1/T$ where $T \sim \text{Gamma}(k, \mu)$ is called an Inverse Gamma distribution with shape parameter k and scale parameter μ (write $1/T \sim \text{InvGamma}(k, \mu)$). Note that what was the rate parameter μ for the

Gamma becomes a scale parameter for the Inverse Gamma. The mean of this $\text{InvGamma}(k, \mu)$ random variable is $\mu/(k - 1)$.

That means a unbiased estimate for μ is

$$\hat{\mu}_{\text{unbiased},n} = \frac{n - 1}{Y_1 + \dots + Y_n}$$

since the right hand side is $\text{InvGamma}(n, (n - 1)\mu)$.

What about discrete variables that are inherently unscalable? In [1], the author presented a method for turning a stream of iid Bernoulli random variables (which are 1 with probability p , and 0 with probability $1 - p$) into a $\text{Gamma}(k, p)$ random variable, where k is a parameter chosen by the user. This could then be used with the known relative error estimate for exponentials to obtain a known relative error estimate for Bernoullis.

While the Bernoulli application has the widest use, Poissons do appear in the output of a Monte Carlo approach to high dimensional integration called the Tootsie Pop Algorithm (TPA) [3, 4]. Therefore, to use TPA to build (ϵ, δ) -approximation algorithms, it is useful to have a known relative error distribution for Poisson random variables.

The remained of this paper is organized as follows. Section 2 describes the new estimate and why it works. It also bounds the expected running time. Section 3 then shows how this procedure can be used together with TPA to obtain (ϵ, δ) -approximations for normalizing constants of distributions.

2 The method

The new estimate is based upon properties of Poisson point processes.

Definition 3. A Poisson point process of rate μ on \mathbb{R} is a random subset $P \subset \mathbb{R}$ such that the following holds.

- For all $a \leq b$, $\mathbb{E}[\#(P \cap [a, b])] = \mu(b - a)$.
- For all $a \leq b \leq c \leq d$, $\#(P \cap [a, b])$ and $\#(P \cap [c, d])$ are independent.

It is well known that there are (at least) two ways to construct a Poisson point process, which forms the basis of the estimate.

The first method for simulating a Poisson point process is to take advantage of the fact that the number of points within a given interval has a Poisson distribution.

Fact 4. Let P be a Poisson point process of rate μ . Then for all $a \leq b$, $\#(P \cap [a, b]) \sim \text{Pois}(\mu(b - a))$. Moreover, conditioned on the number of points in the interval, the points themselves are uniformly distributed over the interval. That is,

$$[P \cap [a, b] | \#(P \cap [a, b]) = n] \sim \text{Unif}([a, b]^n).$$

The second method to building a Poisson point process of rate μ is to use the fact that the distances between successive points are iid exponentially distributed with rate μ .

Fact 5. Let P be a Poisson point process of rate μ . Also, let $P \cap [0, \infty) = \{P_1, P_2, \dots\}$ where $P_i \leq P_{i+1}$ for all i . Setting $A_i = P_{i+1} - P_i$ (and $A_1 = P_1$), we have that $A_1, A_2, \dots \stackrel{iid}{\sim} \text{Exp}(\mu)$.

Using Fact 3, P_k will have a Gamma distribution with shape parameter k and rate parameter μ . So, this is how the estimate works. First, generate N_1 , the number of points of the Poisson point process in $[0, 1]$. If this is at least k , then we know that $P_k \in [0, 1]$. Otherwise, generate N_2 , the number of points in $[1, 2]$. If $N_1 < k$ and $N_1 + N_2 \geq k$, then $P_k \in [0, 2]$. Otherwise, keep going, generating more Poisson random variates until we know that $P_k \in [i, i + 1]$ for some integer i .

Let $A = N_1 + \dots + N_{i-1}$. Then we know that $A < k$ points are in $[0, i]$, and $A + N_i \geq k$. From Fact 4, the N_i points are uniformly distributed over $[i, i + 1]$. The $k - A$ smallest of these points will be P_k . One more well known fact about the order statistics of uniform random variables will be helpful.

Fact 6. If $U_1, \dots, U_n \stackrel{iid}{\sim} \text{Unif}([0, 1])$, then $U_{(i)} \sim \text{Beta}(i, n - i + 1)$.

Putting this all together gives the following estimate, called the Gamma Poisson Approximation Scheme, or GPAS for short.

GAMMA_POISSON_APPROXIMATION_SCHEME	
<i>Input:</i> k <i>Output:</i> $\hat{\mu}_k$	
<hr/>	
1)	$A \leftarrow 0, i \leftarrow 0$
2)	While $A < k$ [Draw k points.]
3)	$T \leftarrow \text{Pois}(\mu)$
4)	If $A + T \geq k$ [Then have k points.]
5)	$T' \leftarrow i + \text{Beta}(k - A, T - (k - A) + 1)$
6)	$A \leftarrow A + T, i \leftarrow i + 1$
7)	$\hat{\mu}_k \leftarrow (k - 1)/T'$

Lemma 1. *The expected number of Poisson random variables drawn by GPAS is bounded above by $1 + k/\mu$.*

Proof. The number of Poisson random variables drawn is $\lceil P_k \rceil \leq P_k + 1$. Since $P_k \sim \text{Gamma}(k, \mu)$, $\mathbb{E}[P_k] = k/\mu$, which shows the result. \square

Note that any fixed time algorithm would need a similar number of samples to obtain such a result.

Fact 7. *The Fisher information of μ for $X \sim \text{Pois}(\mu)$ is $1/\mu$.*

Therefore, by the Crámer-Rao inequality, the variance of any unbiased estimate $\hat{\mu}$ that uses n draws is at least μ/n , so for k/μ draws, the standard deviation will be at least μ/\sqrt{k} .

Lemma 2. *The output $\hat{\mu}_k$ of GPAS has distribution $\text{InvGamma}(k, (k-1)\mu)$, and has standard deviation $\mu/\sqrt{k-2}$.*

Therefore to first order for the same number of samples, the resulting unbiased estimate achieves the minimum variance. Of course, the real benefit of using GPAS is that it provides an exact relative error distribution, thus allowing for precise calculations of the chance of error.

Example 1. For $k = 1000$, GPAS is a $(0.1, 0.0018)$ -approximation algorithm for μ .

$$\begin{aligned} \mathbb{P}((1-0.1)\mu \leq 999/T' \leq (1+0.1)\mu) &= \mathbb{P}(999/0.9 \geq T'/\mu \geq 999/1.1) \\ &= 0.001786\dots \end{aligned}$$

since $T'\mu \sim \text{Gamma}(1000, 999)$.

Example 2. What should k be in order to make GPAS an $(0.1, 10^{-6})$ -approximation algorithm?

Increasing the value of k in the previous example until we reach the first place where $\mathbb{P}((k-1)/0.9 \geq T'/\mu \geq (k-1)/1.1)$ gives $k = 2561$ as the first place where this occurs.

In fact, in the previous example

$$\mathbb{P}(2560/0.9 \geq T'/\mu \geq 2560/1.1) = 0.0000009970\dots,$$

and so is slightly smaller than the error bound requested. It is possible to create an algorithm with exactly 10^{-6} chance of failure by running GPAS either with $k = 2561$ or $k = 2560$ with the appropriate probabilities. This gives the following algorithm, where p_k is the cumulative distribution function of a Gamma distribution with shape k and rate $k - 1$.

EXACT_GPAS

Input: ϵ, δ *Output:* $\hat{\mu}$

- 1) Let $f_i(s) = q_i(1/(1 + \epsilon)) + (1 - q_i(1/(1 - \epsilon)))$
 - 2) Let $k \leftarrow \min\{i : f_i(s) \leq \delta\}$
 - 3) $p \leftarrow (\delta - f_k(s))/(f_{k-1}(s) - f_k(s))$
 - 4) Draw $C \leftarrow \text{Bern}(p)$
 - 5) If $C = 1$ then $k \leftarrow k - 1$
 - 6) $\hat{\mu} \leftarrow \text{GAMMA_POISSON_APPROXIMATION_SCHEME}(k)$
-

3 Applications

So why approximate the mean of a Poisson in the first place? One of the applications is to the Tootsie Pop Algorithm (TPA) [3, 4]. Given a set $A \subset B \in \mathbb{R}^n$, the purpose of TPA is to estimate $\nu(B)/\nu(A)$ for some measure ν .

This is exactly the problem of approximating a high dimensional integral that arises in such problems as finding the normalizing constant of a posterior distribution in Bayesian applications. The output of TPA (see [3, 4]) is exactly a Poisson random variable with mean $\ln(\nu(B)/\nu(A))$.

Typically the situation is that $\nu(A)$ is known, and the goal is to approximate the other. Let $r = \ln(\nu(B)/\nu(A))$. Then if \hat{r} is an approximation for r , then $\exp(\hat{r})$ is an approximation for $\nu(B)/\nu(A)$, and $\nu(A) \exp(\hat{r})$ is an approximation for $\nu(B)$.

An (ϵ, δ) -approximation for $\nu(B)$ can therefore be obtained by finding an (ϵ, δ) -approximation for $\exp(r)$. Note

$$\begin{aligned} \mathbb{P}((1 - \epsilon)e^r \leq \exp(\hat{r}) \leq (1 + \epsilon)e^r) &= \mathbb{P}(r + \ln(1 - \epsilon) \leq \hat{r} \leq r + \ln(1 + \epsilon)) \\ &= \mathbb{P}\left(1 + \frac{\ln(1 - \epsilon)}{r} \leq \frac{\hat{r}}{r} \leq 1 + \frac{\ln(1 + \epsilon)}{r}\right). \end{aligned}$$

Since $|\ln(1 + \epsilon)| < |\ln(1 - \epsilon)|$, the needed bound on the relative error is $\ln(1 + \epsilon)/r$.

A two-phase procedure is used to obtain the estimate. In the first phase, r is estimated with a $(\epsilon, \delta/2)$ -approximation called \hat{r}_1 . So with probability at least $1 - \delta/2$, it holds that $r \geq \hat{r}_1/(1 - \epsilon)$. In the second phase, r is estimated with a $(\ln(1 + \epsilon)\hat{r}_1^{-1}(1 - \epsilon), \delta/2)$ -approximation called \hat{r}_2 .

Using the union bound, the chance that both phases are successful is at least $1 - \delta/2 - \delta/2 = 1 - \delta$, and the above calculation shows that $\exp(\hat{r}_2)$ is an (ϵ, δ) -approximation for $\exp(r)$. The resulting algorithm can be given as follows.

TPA_APPROXIMATION_SCHEME	
<i>Input:</i> ϵ, δ	<i>Output:</i> $\hat{\nu}(B)/\nu(A)$
1) $\hat{r}_1 \leftarrow \text{EXACT_GPAS}(\epsilon, \delta/2)$ 2) $\hat{r}_2 \leftarrow \text{EXACT_GPAS}(\ln(1 + \epsilon)\hat{r}_1^{-1}(1 - \epsilon), \delta/2)$ 3) Output $\exp(\hat{r})$	

This algorithm applies with the understanding that line 3 of the algorithm GAMMA_POISSON_APPROXIMATION_SCHEME is replaced with $T \leftarrow \text{TPA}$, that is, the Poisson with mean μ is replaced by a call to TPA.

ϵ	δ	$\mathbb{E}[T]$ for new method	$\mathbb{E}[T]$ from older method in [4]
0.2	0.2	607 ± 5	1205
0.2	0.01	1753 ± 8	2773
0.1	0.01	5420 ± 10	8415

Table 1: The expected number of calls to TPA for given (ϵ, δ) . Based off of 1000 simulations. Times reported as mean of sample plus or minus standard deviation of sample.

Table 1 shows the expected running time for the new algorithm versus the old, which used Chernoff inequalities to bound the tails of the Poisson distribution. The improvements are in the second order, which is why as δ shrinks relative to ϵ , the improvement is lessened. Still, for reasonable values of (ϵ, δ) , the improvement is very noticeable.

Example 3. Consider the Ising model [5], where each node of a graph with vertex set V and edge set E is assigned either a 0 or 1. For a configuration

$x \in \{0, 1\}^V$, let $H(x) = \#\{e = \{i, j\} \in E : x(i) = x(j)\}$. Then say that X is a draw from the Ising model if $\mathbb{P}(X = x) = \exp(\beta H(x))/Z(\beta)$, where $Z(\beta) = \sum_{y \in \{0, 1\}^V} \exp(\beta H(y))$ is known as the *partition function*

The goal is to find the partition function for various values of β . Note that $Z(0) = 2^{\#V}$ is known, so finding $Z(\beta)/Z(0)$ is sufficient to find $Z(\beta)$.

Considering the Ising model on the 4×4 square lattice with 16 nodes in order to keep the numbers reasonable. Then $Z(1) \approx 3.219 \cdot 10^{11}$ and $\ln(Z(1)/Z(0)) \approx 15.40$. The method for using TPA on a Gibbs distribution is found on p. 99 of [4]. Methods for generating samples from the Ising model for use in TPA abound. See for instance [7, 6, 9, 2]. As long as β is not too high, these methods are very fast.

Using 100 calls with $(\epsilon, \delta) = (0.2, 0.01)$ gives an estimate of 5200 ± 70 for the number of calls needed with the new Poisson estimate, while the old method requires 23249, making the new approach over 4 times as fast in this instance for the same error guarantee.

References

- [1] M. Huber. A Bernoulli mean estimate with known relative error distribution. *Random Structures Algorithms*. arXiv:1309.5413. To appear.
- [2] M. L. Huber. A bounding chain for Swendsen-Wang. *Random Structures Algorithms*, 22(1):43–59, 2003.
- [3] M. L. Huber and S. Schott. Using TPA for Bayesian inference. *Bayesian Statistics 9*, pages 257–282, 2010.
- [4] M. L. Huber and S. Schott. Random construction of interpolating sets for high dimensional integration. *Journal of Applied Probability*, 51(1):92–105, 2014. arXiv:1112.3692.
- [5] E. Ising. Beitrag zur theorie des ferromagnetismus. *Z. Phys.*, 31:253–258, 1925.
- [6] A. Mira, J. Møller, and G.O. Roberts. Perfect slice samplers. *J. R. Statist. Soc. Ser. B Stat. Methodol.*, 63:593–606, 2001.
- [7] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures Algorithms*, 9(1–2):223–252, 1996.

- [8] K. M. Ramachandran and C. P. Tsokos. *Mathematics Statistics with Applications*. Elsevier Academic Press, 2009.
- [9] R. Swendsen and J-S. Wang. Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Let.*, 57:2607–2609, 1986.